

IN THE CLAIMS

What is claimed is:

Sub  
B1

1 1. A communication system comprising:  
2 at least two functional blocks, wherein a first functional block  
3 communicates with a second functional block by establishing a connection,  
4 wherein a connection is a logical state in which data may pass between the  
5 first functional block and the second functional block; and  
6 a bus coupled to each of the functional blocks and configured to carry a  
7 plurality of signals, wherein the plurality of signals comprises a connection  
8 identifier that indicates a particular connection that a data transfer is part of.

1 2. The communication system of claim 1, wherein the plurality of  
2 signals further comprises a thread identifier that indicates a transaction  
3 ~~stream that the data transfer is part of.~~

1 3. The communication system of claim 2, further comprising:  
2 an initiator functional block that sends transfer requests;  
3 an initiator interface module coupled to the initiator functional block  
4 and to the bus;  
5 a target functional block that responds to transfer requests; and  
6 a target interface module coupled to the target functional block,  
7 wherein the connection identifier is sent with a transfer request from the  
8 initiator interface module to the target interface module.

sub  
a2  
1 4. The communication system of claim 3, wherein the thread  
2 identifier is sent from the target interface module to the target functional  
3 block and from the initiator interface module to the initiator functional block.

1 5. The communication system of claim 4, wherein the connection  
2 identifier is sent from the target interface module to the target functional  
3 block and from the initiator interface module to the initiator functional block.

1 6. The communication system of claim 5 wherein the connection  
2 identifier is a multi-bit value that encodes information including:  
3 a transfer priority;  
4 a transfer order; and  
5 an functional block that originated the transfer.

sub  
B2  
1 7. The communication system of claim 6, wherein the connection  
2 identifier is one of a plurality of connection identifiers associated with an  
3 initiator functional block, and wherein the connection identifier is mapped to  
4 a thread identifier by the initiator interface module.

5  
8. 4 The communication system of claim 7, wherein the connection  
2 identifier is one of a plurality of connection identifiers associated with a target  
3 functional block that supports simultaneous connections, and wherein the  
4 target functional block acquires a connection identifier of an open connection  
5 and maps the connection identifier to a thread identifier.

sub  
a4  
1 9. The communication system of claim 2, wherein the plurality of  
2 signals further comprises:

3 a request thread ID signal that indicates a thread number associated  
4 with a current transaction intended for a target functional block;  
5 a response thread ID signal that indicates a thread that a transfer from  
6 the target functional block is part of;  
7 a request thread busy signal that indicates that indicates to an initiator  
8 functional block that the target functional block cannot receive new requests  
9 associated with certain threads; and  
10 a response thread busy signal that indicates that the initiator functional  
11 block cannot receive any new responses from the target functional block that  
12 ~~are associated with certain threads.~~

1 10. A method for communicating between functional blocks in a  
2 computer system, the method comprising the steps of:  
3 establishing a plurality of connection identifiers, wherein each  
4 connection identifier associates a particular data transfer with a particular  
5 connection, wherein a connection is a logical state in which data may pass  
6 between an initiator functional block of the plurality of functional blocks and  
7 a target functional block of the plurality of functional blocks, and wherein a  
8 connection is established when a particular data transfer is initiated; and  
9 allowing an initiator functional block to issue a first transfer "Y" if the  
10 transfer "Y" is an oldest, non-issued, non-retired transfer among a set of  
11 transfer requests with a same connection identifier as the transfer "Y".

1 11. The method of claim 10, further comprising the step of allowing  
2 the initiator functional block to issue the transfer "Y" if every non-retired  
3 transfer with the same connection identifier is older than the transfer "Y" and  
4 is currently issued to a same target functional block as the transfer "Y".

1 12. The method of claim 11, wherein if the transfer "Y" is issued, the  
2 transfer "Y" is considered pipelined with the older, non-retired transfers.

1 13. The method of claim 10, further comprising the step of the target  
2 functional block giving a BUSY response to every later transfer that is  
3 pipelined with the transfer "X" and has a same connection identifier as the  
4 transfer "X" if the target functional block responds to the transfer "X" so that  
5 an initiator initiating the transfer "X" may not retire the transfer "X".

1 <sup>10</sup> 14. The method of claim 13, wherein a transfer "Y" that is issued  
2 after a transfer "X", is older than the transfer "X", and has a same connection  
3 identifier as the transfer "X" is considered not pipelined with the transaction  
4 "X".

1 <sup>12</sup> 15. The method of claim <sup>11</sup> 12, wherein a target functional block  
2 determines whether a transfer is a pipelined transfer based upon when the  
3 transfer occurs and upon a connection identifier associated with the transfer.

1 <sup>13</sup> 16. The method of claim <sup>9</sup> 14, further comprising the steps of:  
2 an initiator functional block maintaining a time-ordered queue of  
3 desired transfers with a same connection identifier;  
4 the initiating functional block marking a transfer as non-issued and  
5 non-retired as it is entered into the queue.

1 <sup>14</sup> 17. The method of claim <sup>13</sup> 16, further comprising the steps of:

2 if a next oldest entry is non-retired and addresses a same target  
3 functional block, marking the transfer as pipelined; else  
4 marking the transfer as non-pipelined.

1 <sup>15</sup>  
~~18.~~ The method of claim <sup>14</sup>~~17~~, further comprising the step of, when a  
2 transfer issues, marking the transfer as issued.

1 <sup>16</sup>  
~~19.~~ The method of claim <sup>15</sup>~~18~~, further comprising the step of, when a  
2 transfer is completed, marking the transfer as non-issued.

1 <sup>17</sup>  
~~20.~~ The method of claim <sup>16</sup>~~19~~, further comprising the step of, if the  
2 transfer is successfully completed, marking the transfer as retired; and  
3 deleting the transfer from the queue.

1 <sup>18</sup>  
~~21.~~ The method of claim <sup>17</sup>~~20~~, further comprising the step of, if the  
2 transfer is not successfully completed, re-attempting the transfer.

1 <sup>19</sup>  
~~22.~~ The method of claim <sup>9</sup>~~14~~, further comprising the step of the target  
2 functional block maintaining a time-ordered queue having a depth that is a  
3 number of bus clock cycles between a request for a transfer and a response to  
4 the request.

1 <sup>20</sup>  
~~23.~~ The method of claim <sup>19</sup>~~22~~, further comprising the steps of:  
2 on each cycle of the bus clock, retiring an oldest entry in the time-  
3 ordered queue; and

4 on each cycle of the bus clock, adding a new entry to the time-ordered  
5 queue, including a connection identifier associated with a current request for  
6 a transfer.

1 <sup>21</sup>  
~~24.~~ The method of claim <sup>20</sup>~~23~~, further comprising the steps of:  
2 if a current request for a transfer contains a valid transfer that selects  
3 the target functional block, allowing a FIRST bit and a BUSY bit of an entry in  
4 the time-ordered queue to be set, wherein a set FIRST bit implies that an  
5 associated transfer is a first transfer of a set of potentially pipelined transfers;  
6 else  
7 clearing the FIRST bit and the BUSY bit.

1 <sup>22</sup>  
~~25.~~ The method of claim <sup>21</sup>~~24~~, further comprising the step of setting  
2 the FIRST bit if:  
3 no transfer in the time-ordered queue is earlier than a current transfer,  
4 has a same connection identifier as the current transfer and has an associated  
5 FIRST bit set; and  
6 the current transfer will receive a BUSY response due to a resource  
7 conflict.

1 <sup>23</sup>  
~~26.~~ The method of claim <sup>21</sup>~~24~~, further comprising the step of setting  
2 the BUSY bit if:  
3 the target functional block has a resource conflict; or  
4 an earlier transfer in the time-ordered queue has an associated FIRST  
5 bit set and has a same connection identifier as a current transfer.

1 <sup>24</sup>  
~~27.~~ The method of claim <sup>23</sup>~~26~~, further comprising the step of using a  
2 connection identifier to enforce ordering among transfers.

1 <sup>25</sup>  
~~28.~~ The method of claim <sup>24</sup>~~27~~, further comprising the step of:  
2 in response to a first request for a data transfer issued in a first bus cycle,  
3 the target functional block setting a BUSY bit in a first time-ordered queue  
4 entry, wherein a first connection identifier is associated with the first request;  
5 and  
6 in response to a second request for a data transfer in a next bus cycle  
7 subsequent to the first bus cycle, the target functional block clearing a BUSY  
8 bit in a second time-ordered queue entry and performing an action in  
9 connection with executing the data transfer requested in the second request.

Add A7